Codon Evolution
Mechanisms and Models

EDITED BY

Gina M. Cannarozzi
University of Bern, Switzerland

Adrian Schneider
University of Utrecht, The Netherlands
CHAPTER 9

Simulation of coding sequence evolution

Miguel Arenas and David Posada

9.1 Introduction

Computer simulations try to mimic the processes that happen in the real world (Peck, 2004) and are widely used in a large number of different fields. In general, computer simulations allow to study complex systems, including those analytically intractable. Indeed, computer simulations can be considered as experimental systems (Winsberg, 2003), and using appropriate models, they can be more efficient than simple analytic methods in order to gain information about the system. In silico simulations can incorporate stochasticity very easily, which is a typical and inherent property in most biological processes. In molecular evolution, computer simulations have been commonly used to understand the interactions among evolutionary processes, for hypothesis testing, to evaluate and compare different analytical methods, or to estimate evolutionary parameters. Indeed, in order to obtain meaningful biological inferences from simulated data, it is important that the generating models are as realistic as possible. In this chapter, we describe simulation algorithms for the evolution of coding sequences under different evolutionary scenarios, potential uses of these simulations, and current simulation software.

9.2 Simulation of coding sequences

There are different simulation strategies that can be more or less appropriate depending on the question of interest or the purpose of the study. Below the species level, samples of codon sequences can be taken from whole populations simulated forward in time, generation by generation, starting from the sequence of their most recent common ancestor (MRCA). On the other hand, a sample of sequences can be simulated if their genealogy (below the species level) or phylogeny (above the species level) is known. Coalescent theory provides a versatile framework to simulate the genealogy of a sample backward in time. Phylogenies can be given from previous studies, estimated from data, or simulated itself under branching processes like the coalescent or the birth–death process. It is also possible to simulate gene trees within population or species trees. In general, ‘forward’ simulations are extremely flexible, allowing for the simulation of almost any population genetic model. On the other hand, ‘backward’ approaches are computationally much more efficient, especially for short sequences, but have more limitations especially regarding the simulation of selective scenarios and large recombination rates. Both approaches have different advantages and disadvantages (see below) and, in general, they can be considered to complement each other.

9.2.1 Forward simulations

Forward-time simulations describe the evolutionary history of a whole population from the past to the present. Traditionally they were constrained by the available computational power, but in recent years there have been important advances in this regard, such as the implementation of more computationally efficient algorithms (e.g. Peng and Kimmel, 2005; Padhukasahasram et al., 2008). Forward simulations often start from an initial population, which constitutes the basis for the descendant generation. Often, generations are considered discrete
and non-overlapping. After introducing mutations at random, individuals contributing to the next generation are randomly chosen, according to their relative fitness. For sexual diploid populations, recombining gametes are produced and mating pairs are identified in order to generate descendants. In addition, migration, demographic, and speciation events or population history can be considered as well. The process continues for a given number of generations and stops at some arbitrary point from which samples are taken. Currently, we are only aware of two programs that allow for the simulation of coding sequences forward in time: GenomePop (Carvajal-Rodriguez, 2008) and SFS_CODE (Hernandez, 2008).

### 9.2.3 Simulation of codon substitutions

Codon substitutions are usually assumed to be independent among different sites. In forward approaches, coding regions can be easily simulated, defining an open reading frame and assuming given genetic code. In this case, particular mutations are introduced every generation, resulting in synonymous and nonsynonymous changes. When the simulation occurs along lineages, i.e. a coalescent genealogy or a phylogenetic tree, continuous-time Markov models of codon substitution are used instead (see Chapter 1), defined by a $61 \times 61$ matrix $Q$ of instantaneous substitution rates (stop codons are ignored). These codon models define the probability of changing from codon $i$ to codon $j$ along a branch of length $t$, where $t$ is the product of time and substitution rate.

The most popular method to simulate coding sequences using Markov codon models consists of evolving each codon in turn over a rooted tree. For a given site, the codon at the root is usually specified according to the equilibrium frequencies of the chosen model (or directly specified by the user). This codon is evolved towards the tip of the tree, one branch at a time, according to the transition probabilities ($P(t) = e^{Qt}$). This strategy is very flexible and allows for heterogeneous processes operating at different sites or in different parts of the tree. This is the method implemented in programs like EVOLVER in the PAML package (Yang, 1997), SeqGen (Rambaut and Grassly, 1997), and Recodon and NetRecodon.

Another possibility is to use the exponentially distributed waiting times from the Markov chain (Yang, 2006). Starting from the root, the waiting time to the next substitution event can be calculated according to an exponential distribution whose mean is the inverse of the instantaneous rate matrix for the given site. If the waiting time is longer than the length of the branch, there will be no substitution. Otherwise, sampling from to the jump process of the continuous Markov chain (also called the embedded Markov chain), a transition from state $i$ to state $j$ will occur with probability $q_{ij}/q_i$ (Yang, 2006) (where $q_i$ is the substitution rate for nucleotide $i$), and a new substitution will be attempted for the remaining branch length.

This approach has several advantages, as it can be
128 SIMULATION OF CODING SEQUENCE EVOLUTION

Box 9.1 The coalescent with intracodon recombination

Simulation of coding sequences with recombination can be especially useful to mimic the evolution of fast-evolving pathogens, like viruses and bacteria, where the recombination rates can be very high. In the past, the coalescent simulation of recombining sequences only allowed for recombination breakpoints between the substitution units, so in the case of codon models, recombination was forced to occur between codons (Anisimova et al., 2003). This problem was solved by Arenas and Posada (2010a) who recently proposed a new coalescent algorithm that allows for intracodon recombination, implemented in the program NetRecodon (http://darwin.uvigo.es/, software section). In this case, not only the codons that reach the sample (sampled ancestral material), but also the codons that contain sites that evolved at some point with ancestral sites (unsampled ancestral material), need to be tracked. In this way, an ancestral codon graph (ACG), containing sampled and unsampled ancestral material, represents the evolutionary history at each codon. Second, for the simulation along the ACGs, a recursion was proposed to solve the loops introduced by intracodon recombination, which basically waits for pairs of parental nodes to be occupied before proceeding further towards the present (Figure 9.A.1).

applied to the whole sequence, it does not require the calculation of the P matrix and allows for site and context-dependent simulation of insertions and deletions (Fletcher and Yang, 2009). Conveniently, because the embedded Markov chain provides the conditional probabilities of transitioning from state $i$ into state $j$ in one step or generation, the approach suggested by Yang can be used to generate data using codon models in forward simulations (Carvajal-Rodriguez, 2008).

9.3 Uses of simulated coding data

Truth in molecular evolution is usually unknown and simulated data are needed to test the accuracy and robustness of the different methods used to make evolutionary inferences from codon data. Indeed, simulations have been used to evaluate different estimators of the nonsynonymous/synonymous rate ratio ($\omega$), a central parameter of codon models (e.g. Kosakovsky Pond and Frost, 2005; Nickel et al., 2008), or its relationship with the selection coefficient at the intra-specific level (Kryazhimskiy and Plotkin, 2008). Most estimators of $\omega$ do not consider the potentially misleading effect of recombination (Wilson and McVean, 2006), and simulations have been very useful to assess this effect and its causes. In general, different studies have shown that recombination does not have an effect on the estimation of the global $\omega$. 

---

**Figure 9.A.1** An example of codon evolution along an ancestral codon graph with an intracodon recombination event. White and grey circles correspond to coalescence and parental nodes, respectively. Starting from the GMRCA, the codon is evolved between nodes by a recursive function, with substitutions introduced according to the probabilities specified by the codon model and the branch length. The recursion is interrupted if a recombinant node is reached and its corresponding parental node has not been assigned a codon yet; in this case the recursion continues in other direction (note the ‘stop’ in step 3 and continuation in step 4). Only when the two parental nodes have already been assigned a codon, the algorithm combines their codons according to the recombination breakpoint (step 6). The resulting recombinant codon (TTT) continues evolving towards the present (step 7).
<table>
<thead>
<tr>
<th>Program</th>
<th>Category</th>
<th>Substitution model</th>
<th>Rate variation among codons</th>
<th>$\bar{\gamma}$ variation</th>
<th>Indels</th>
<th>Evolutionary scenarios</th>
<th>OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>EVOLVER</td>
<td>Phylo</td>
<td>GY94</td>
<td>No</td>
<td>Sites</td>
<td>No</td>
<td>--</td>
<td>All</td>
</tr>
<tr>
<td>Seq-Gen</td>
<td>Phylo</td>
<td>Codon position</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>R</td>
<td>All</td>
</tr>
<tr>
<td>EvolveAGene3</td>
<td>Phylo</td>
<td>E. coli spectra</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>R, S, P, M</td>
<td>Source code</td>
</tr>
<tr>
<td>INDELible</td>
<td>Phylo</td>
<td>GY94</td>
<td>No</td>
<td>Sites</td>
<td>Yes</td>
<td>R</td>
<td>All</td>
</tr>
<tr>
<td>indel-Seq-Gen vs 2.0</td>
<td>Phylo</td>
<td>Codon position</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>--</td>
<td>All</td>
</tr>
<tr>
<td>SISSI</td>
<td>Phylo</td>
<td>Codon position</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>R</td>
<td>All</td>
</tr>
<tr>
<td>HyPhy</td>
<td>Phylo</td>
<td>GY94 and MG94</td>
<td>$G + I$</td>
<td>Sites</td>
<td>No</td>
<td>R</td>
<td>All</td>
</tr>
<tr>
<td>EvolSimulator</td>
<td>Phylo</td>
<td>Nucleotide</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>R, S, P, M</td>
<td>Source code</td>
</tr>
<tr>
<td>SIMGRAM</td>
<td>Phylo</td>
<td>ECM</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>S</td>
<td>Source code</td>
</tr>
<tr>
<td>SIMGENOME</td>
<td>Phylo</td>
<td>ECM and CodonPAM</td>
<td>No</td>
<td>Sites</td>
<td>Yes</td>
<td>R</td>
<td>Linux</td>
</tr>
<tr>
<td>ALF</td>
<td>Phylo</td>
<td>ECM and CodonPAM</td>
<td>No</td>
<td>Sites</td>
<td>Yes</td>
<td>R, S, D, P, M</td>
<td>Source code</td>
</tr>
<tr>
<td>GenomePop</td>
<td>Forw</td>
<td>MG94</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>R, S, D, P, M, L</td>
<td>All</td>
</tr>
<tr>
<td>SFS_CODE</td>
<td>Forw</td>
<td>Nucleotide</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>R, S, D, P, M, L</td>
<td>Source code +</td>
</tr>
<tr>
<td>CodonRecSim</td>
<td>Coal</td>
<td>GY94</td>
<td>No</td>
<td>Sites</td>
<td>No</td>
<td>R</td>
<td>Source code +</td>
</tr>
<tr>
<td>Recodon</td>
<td>Coal</td>
<td>GY94</td>
<td>$G + I$</td>
<td>Sites</td>
<td>No</td>
<td>R, D, P, M, L</td>
<td>All</td>
</tr>
<tr>
<td>NetRecodon</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

"Category" includes forward (Forw), phylogenetic (Phylo), and coalescent (Coal) approaches. "Substitution model" indicates how substitution events are modelled: "Codon position" indicates that a different substitution rate is specified for each codon position; "Nucleotide" indicates that mutations are filtered afterwards according to predefined fitness effects, "ECM" means empirical codon model; and "CodonPAM" means codon PAM distance. "Rate variation among codons" indicates whether different codons are allowed to evolve under different synonymous rates ($G$: gamma distribution; $I$: proportion of invariable sites). "$\bar{\gamma}$ variation" specifies whether $\bar{\gamma}$ can change across codons and/or branches. "Indels" indicates whether the program considers insertion and deletion events. "Evolutionary scenarios" indicates the possibility of simulation under different evolutionary scenarios that can include recombination 'R' (considering different strategies), selection pressures 'S', demographics 'D', population history 'P', migration 'M', and longitudinal sampling 'L'. "OS" indicates the availability of executable files and/or source code for different operative systems ("All" indicates that Macintosh, Windows, and Linux executables are available).

1 $G + I$ could be available in a newer version (current is 1.03; personal communication).
2 If codons are treated as independent loci.
3 Only intercodon recombination.
for a given alignment (e.g. Anisimova et al., 2003; Kosakovsky Pond et al., 2008; but see Shriner et al., 2003). Our own simulations indicate that this is also the case when $\omega$ is constant along the alignment but varies among lineages (unpublished data). Importantly, recombination can also result in the generation of false positively selected sites and/or spurious inference of selection when likelihood-ratio tests (LRTs) are used (Anisimova et al., 2003; Shriner et al., 2003; Arenas and Posada, 2010a). On the other hand, recombination can seriously bias the reconstruction of ancestral sequences, in particular when dealing with coding sequences (Arenas and Posada, 2010b).

### 9.4 Software implementations

Different computer programs have been developed for the simulation of coding sequences. Tables 9.1 and 9.2 show the most relevant programs and their main characteristics and possibilities, like their general approach (e.g. backward), substitution model, possibility of synonymous and nonsynonymous rate heterogeneity, ability to simulate indels, possible evolutionary scenarios, and operative system. Only a few of these programs use a forward approach. GenomePop (Carvajal-Rodriguez, 2008) implements recombination, population structure and demographic, diploid/haploid models, and fitness-based selection. In addition, it implements complex codon models, like GTR × MG94. A very friendly forward simulator is SFS_CODE (Hernandez, 2008), which implements a wide variety of mutation models, including indels and many different scenarios, such as population expansions and bottlenecks, multiple populations, distribution of selective effects, and consideration of recombination hotspots for crossing-over and gene conversion events.

On the other hand, most codon simulators evolve the sequences along phylogenetic trees. EVOLVER from the PAML package (Yang, 1997) was one of the first programs capable of generating coding data. More recently, INDELible (Fletcher and Yang, 2009) implements complex substitution models that include indels, and $\omega$ rate variation among sites and/or branches. The program indel-Seq-Gen 2.0 (Strope et al., 2009), an extension of Seq-Gen (Rambaut and Grassly, 1997), allows simulation under a wide variety of indel models and includes other options like pseudogene simulation or motif conservation. EvolveAGene (Hall, 2005; 2008) evolves a real coding sequence along a given tree using a mutation spectrum experimentally determined from *Escherichia coli*, and including indels. Other tools like SISSI (Gesell and Von Haeseler, 2006) implement the simulation of codons with structural dependency among sites. The program HyPhy (Kosakovsky Pond et al., 2005) implements a wide range of codon models that can be suited to particular user requirements using its flexible scripting language.

Among the coalescent-based programs CodonRecSim (Anisimova et al., 2003) and Recodon can simulate coding sequences with recombination but forcing breakpoints to occur between codons. This stringent assumption was avoided in NetRe-

### Table 9.2 URLs for the different coding-sequence simulators

<table>
<thead>
<tr>
<th>Program</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>EVOLVER</td>
<td><a href="http://abacusgene.ucl.ac.uk/software/paml.html">http://abacusgene.ucl.ac.uk/software/paml.html</a></td>
</tr>
<tr>
<td>Seq-Gen</td>
<td><a href="http://tree.bio.ed.ac.uk/software/seggen/">http://tree.bio.ed.ac.uk/software/seggen/</a></td>
</tr>
<tr>
<td>EvolveAGene3</td>
<td><a href="http://web.me.com/barryghall/Software/Software.html">http://web.me.com/barryghall/Software/Software.html</a></td>
</tr>
<tr>
<td>INDELible</td>
<td><a href="http://abacusgene.ucl.ac.uk/software/indelible/">http://abacusgene.ucl.ac.uk/software/indelible/</a></td>
</tr>
<tr>
<td>indel-Seq-Gen vs 2.0</td>
<td><a href="http://rec.uga.edu/software/uppgisg/">http://rec.uga.edu/software/uppgisg/</a></td>
</tr>
<tr>
<td>SISSI</td>
<td><a href="http://www.cbiv.ait/software/sissi/">http://www.cbiv.ait/software/sissi/</a></td>
</tr>
<tr>
<td>SIMGRAM</td>
<td><a href="http://biowiki.org/SimulationTools">http://biowiki.org/SimulationTools</a></td>
</tr>
<tr>
<td>/SIMGENOME</td>
<td></td>
</tr>
<tr>
<td>ALF</td>
<td><a href="http://www.cbrg.ethz.ch/alf">http://www.cbrg.ethz.ch/alf</a></td>
</tr>
<tr>
<td>GenomePop</td>
<td><a href="http://webs.uvigo.es/acraaj/GenomePop.htm">http://webs.uvigo.es/acraaj/GenomePop.htm</a></td>
</tr>
<tr>
<td>SFS_CODE</td>
<td><a href="http://sfscode.sourceforge.net/SFS_CODE_home.html">http://sfscode.sourceforge.net/SFS_CODE_home.html</a></td>
</tr>
<tr>
<td>CodonRecSim</td>
<td><a href="http://fisher.berkeley.edu/cleg/software.html">http://fisher.berkeley.edu/cleg/software.html</a></td>
</tr>
<tr>
<td>Recodon</td>
<td><a href="http://people.binf.ku.dk/rasmus/webpage/CodonRecSim.html">http://people.binf.ku.dk/rasmus/webpage/CodonRecSim.html</a></td>
</tr>
<tr>
<td>NetRecodon</td>
<td><a href="http://darwin.uvigo.es/">http://darwin.uvigo.es/</a></td>
</tr>
</tbody>
</table>

All websites last accessed September 2011.
codon, where coding sequences are evolved on the ancestral recombination graph using a more sophisticated algorithm.

At the genomic scale, EvolSimulator (Beiko and Charlebois, 2007) simulates the evolution of gene families under non-stationary models, including lateral transfer and gene gains and losses. The suite GSIMULATOR/SIMGRAM/SIMGENOME (Varadarajan et al., 2008) can simulate coding genomic regions using models trained in the PANDIT database (Whelan et al., 2003). Indeed, some programming libraries offer some functionalities to evolve codon sequences, like Darwin (Gonnet et al., 2000) and Bio++ (Dutheil and Boussau, 2008). Finally, A recent package for simulating genome evolution is ALF (Dalquen et al., in press), which models evolution both at gene and genome level. It can simulate coding sequences according to parametric M-series codon models (Yang et al., 2000) as well as the empirical codon models ECM (Kosiol et al., 2007) and CodonPAM (Schneider et al., 2005). In addition, it has various other options (see Table 9.1). The application is already written in Darwin.

Acknowledgements

This work was funded by the Spanish Ministry of Education and Science (MEC) through an FPI fellowship BES-2005-9151 to M.A., and a research grant BIO2007-61411 to D.P. We thank Gina Caninarozzi and an anonymous reviewer for their comments.

References


selection pressures on HIV-1 using phylogenetic likelihood models. Statistics in Medicine, 27, 4779–4789.